



ICS254 (Discrete Structures II))

Term: 172

Implementation of the Diffie-Hellman Protocol Programming Assignment Report

Group #6

Doctor in Charge:

Dr. Wasfi AlKhteeb

Group Members:

Name	ID
Yousef Majeed	201568070
Thamer Mashni	201417240

Saturday, April 28, 2018

“Group #07”[®] - *“Success is a journey not a destination!”*

Table of Contents

1. The objective(s) of the protocol	3
2. How and under which restrictions/environment does it work?	3
3. Public Information and Private Information	3
3.1 Public	3
3.2 Private	3
4. References	4
5. How to compile and run the code	4
6. Specific details of the algorithms that we have implemented	4
7. Sample runs of our program	5
8. Work Distribution	5

Implementation Of the Diffie-Hellman Protocol	Version: 1.0
Programming Assignment Report	Date: April 28, 2018

Implementation of the Diffie-Hellman Protocol

1. The objective(s) of the protocol

To make two parties able to exchange a secret key over an insecure communications channel without having shared any information in the past.

2. How and under which restrictions/environment does it work?

- A sender and a receiver need to share a common key
- First the sender and the receiver agree to use a prime p and a primitive root a of p .
- Then the sender chooses a secret integer k_1 and sends $a^{k_1} \bmod p$ to the receiver .
- The receiver chooses a secret integer k_2 and sends $a^{k_2} \bmod p$ to the sender
- The sender computes $(a^{k_2})^{k_1} \bmod p$
- The receiver computes $(a^{k_1})^{k_2} \bmod p$
- At the end of this protocol, sender and receiver have computed their shared key, namely $(a^{k_2})^{k_1} \bmod p = (a^{k_1})^{k_2} \bmod p$.

3. Public Information and Private Information

3.1 Information can be made Public:

- a) p
- b) a
- c) $a^{k_1} \bmod p = A$
- d) $a^{k_2} \bmod p = B$

3.2 Information can be made Private:

- a) k_1
- b) k_2
- c) and the common key $(a^{k_2})^{k_1} \bmod p = (a^{k_1})^{k_2} \bmod p$

Implementation Of the Diffie-Hellman Protocol	Version: 1.0
Programming Assignment Report	Date: April 28, 2018

4. References

- Kenneth, H. *Discrete Mathematical and its Applications, Seventh Edition*, course Textbook.

5. How to compile and run the code

We have implemented the application with simple user interface to facilitate working and testing the application as shown in part 7. First you choose the desired operation (using the protocol or cracking protocol), then enter the required input for the operation then click on compute button (or crack) to show the result.

6. Specific details of the algorithms that have been implemented

Regarding to the BONUS part, we have implemented the discrete logarithm algorithm to break the security of the Diffie-Hellman protocol, so the algorithm says:

Suppose that p is a prime, r is a primitive root modulo p , and a is an integer between 1 and $p - 1$ inclusive. If $r^e \bmod p = a$ and $0 \leq e \leq p - 1$, we say that e is the *discrete logarithm* of a modulo p to the base r and we write $\log_r a = e$ (where the prime p is understood).

So, it is clear that it can be worked with our case where the public information will be placed in the equation to find the private keys then after that, calculate the shared secret key will be done.

'p' is the same name of our prime

'r' is the primitive root = 'a' in our implementation

'a' is the public key = 'A' and 'B' in our implementation

'e' is the private key = 'k1' and 'k2' in our implementation

so, the equations will be:

$$a^{k1} \bmod p = A$$

$$a^{k2} \bmod p = B$$

7. Sample runs of the program

Application interface:

home panel:

DiffieHellman

Welcome to Diffie Hellman Protocol

*Please Choose Which Operation You Want to Use?

☒ Calculate a Shared Key

☐ Crack a Shared Key

Proceed

compute DHP panel:

DiffieHellman

Calculate a Shared Key

Enter a Prime Number 'p'

Enter a Primitive Root 'a'

The Shared Key

Cracking DHP panel:

The screenshot shows a window titled "DiffieHellman" with a red heading "Crack a Shared Key". It contains several input fields and buttons:

- Enter a Prime Number 'p'**: An input field with a button "Is it Prime?" next to it.
- Enter a Primitive Root 'a'**: An input field with a button "Is it Primitive Root?" next to it.
- Enter Public Key 'A'**: An input field.
- Enter Public Key 'B'**: An input field.
- Crack**: A button to perform the cracking operation.
- The Shared Key is**: An input field to display the result.
- Show the Private Keys**: A button to reveal the private keys.
- Two empty input fields for private keys.
- Back**: A button to return to the previous screen.

Valid Runs:

a- For compute DHP

1. since $\text{MAX_VALUE} = 2147483647$

p = prime number larger than Integer.MAX_VALUE = 21147483647

a = PRIMITIVE ROOT of $p = 4567$

The screenshot shows a window titled "DiffieHellman" with a red heading "Calculate a Shared Key". It contains several input fields and buttons:

- Enter a Prime Number 'p'**: An input field containing the value "21147483647" with a button "Is it Prime?" next to it.
- Enter a Primitive Root 'a'**: An input field containing the value "4567" with a button "Is it Primitive Root?" next to it.
- Calculate**: A button to perform the calculation.
- The Shared Key**: An input field containing the value "13299566600".
- Show the Private Keys**: A button to reveal the private keys.
- Show the Public Keys**: A button to reveal the public keys.
- Two rows of output fields:
 - Private Key 'k1' is: 3948
 - Public Key 'A' is: 10708986208
 - Private Key 'k2' is: 2527224
 - Public Key 'B' is: 6477825894
- Back**: A button to return to the previous screen.

2. Another valid run with $p = 61173959573$, $a = 7474$

The screenshot shows a window titled "DiffieHellman" with the following elements:

- Title Bar:** DiffieHellman, with standard window controls (minimize, maximize, close).
- Header:** "Calculate a Shared Key" in red text.
- Inputs:**
 - "Enter a Prime Number 'p'" with value 61173959573 and a button "Is it Prime?".
 - "Enter a Primitive Root 'a'" with value 7474 and a button "Is it Primitive Root?".
- Buttons:** "Calculate" (centered).
- Output:** "The Shared Key" with value 34661398843.
- Buttons:** "Show the Private Keys" and "Show the Public Keys".
- Private Keys:**
 - Private Key 'k1' is: 22929605
 - Private Key 'k2' is: 141217
- Public Keys:**
 - Public Key 'A' is: 59940298168
 - Public Key 'B' is: 54193091959
- Buttons:** "Back" (bottom center).

2) For Cracking DHP

1. With $p = 21147483647$ and $a = 4567$ and randomly entered $A = 2345678$ & $B=45678$

The screenshot shows a window titled "DiffieHellman" with the following elements:

- Title Bar:** DiffieHellman, with standard window controls (minimize, maximize, close).
- Header:** "Crack a Shared Key" in red text.
- Inputs:**
 - "Enter a Prime Number 'p'" with value 21147483647 and a button "Is it Prime?".
 - "Enter a Primitive Root 'a'" with value 4567 and a button "Is it Primitive Root?".
 - "Enter Public Key 'A'" with value 2345678.
 - "Enter Public Key 'B'" with value 45678.
- Buttons:** "Crack" (centered).
- Output:** "The Shared Key is" with value 7656435240.
- Buttons:** "Show the Private Keys".
- Private Keys:**
 - Private Key 'k1' is: 2029161097
 - Private Key 'k2' is: 11227557394
- Buttons:** "Back" (bottom center).

2. Another run with $p = 61173959573$ and $a = 7474$ and randomly entered $A = 2345678$ & $B = 45678$

The screenshot shows a Java application window titled "DiffieHellman" with a red heading "Crack a Shared Key". It contains several input fields and buttons:

- Enter a Prime Number 'p'**: Input field with value 61173959573, followed by a button "Is it Prime?".
- Enter a Primitive Root 'a'**: Input field with value 7474, followed by a button "Is it Primitive Root?".
- Enter Public Key 'A'**: Input field with value 2345678.
- Enter Public Key 'B'**: Input field with value 45678.
- A button labeled "Crack".
- The Shared Key is**: Input field with value 16037594621.
- A button labeled "Show the Private Keys".
- Below that, two input fields: "te Key 'k1' is: 27719153679" and "te Key 'k2' is: 59951460265".
- A button labeled "Back" at the bottom.

Invalid Runs:

- a- For compute DHP

1.

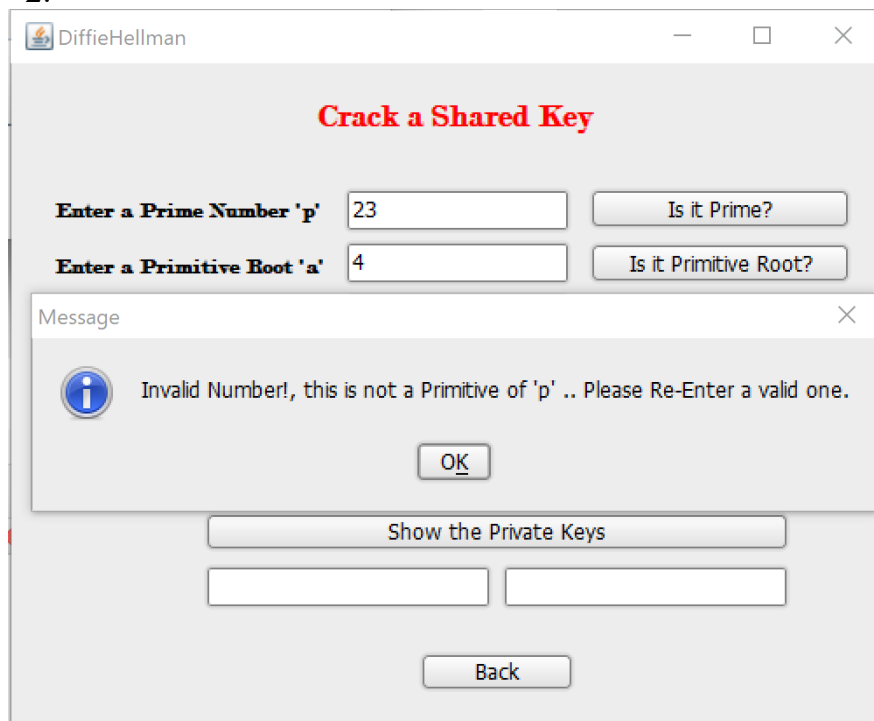
The screenshot shows the "DiffieHellman" application window with a red heading "Calculate a Shared Key". It includes an input field for "Enter a Prime Number 'p'" with the value 909819912 and a button "Is it Prime?". A modal message box is displayed over the main window with the text: "Invalid Number!, this is not a PRIME Number .. Please Re-Enter a valid one." with an "OK" button. Below the message box, there are buttons for "Show the Private Keys" and "Show the Public Keys", followed by two empty input fields, and a "Back" button at the bottom.

2.

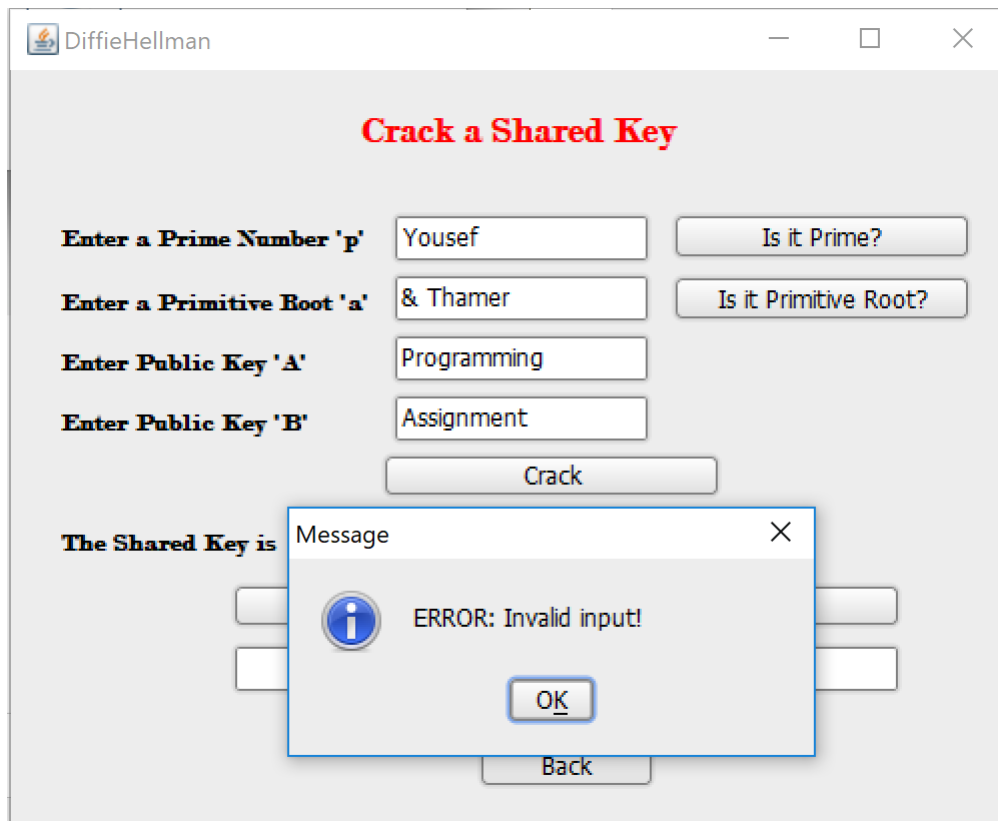
b- For Cracking DHP

1.

2.



c- Invalid input Format



8. Work Distribution

TASK RECORD

Date	Member	Task Details
18 April 2018	Thamer Mashni	Starting the project, do some most initial functions
22 April 2018	Yousef Majeed	Working on the main functionality of the program
24 April 2018	Yousef Majeed	Start working on the Bonus part and close to finish it
25 April 2018	Thamer Mashni	Make the program as a GUI Program
27 April 2018	Yousef Majeed	Enhance the GUI Program a little
28 April 2018	Both Members	Finalize Work and Finalize the Report